Homework 6
Fundamental Algorithms, Fall 2011, Professor Yap

Due: Thu Nov 17, in class.

INSTRUCTIONS:

• Please write clearly, and economically. Use notations we introduce in lecture notes and in lectures.

• If you cannot write clearly, then consider typesetting the solutions (not recommended in general because this takes too much time).

• AGAIN: you get 2 bonus points just for writing your name in the correct format: [LAST-NAME-in-CAPS] [comma] [First-Name]. E.g., ”YAP, Chee K.” or ”YAP, Chee” are good, but ”Yap, Chee” or ”Chee Yap” or ”Yap Chee K.” are no good.

1. (5 Points, Radius and Diameter)
   One of the scratched problems in hw5 is about the radius and diameter of a connected bigraph \( G \). For the current problem, we want you to prove a nice observation of your fellow classmate, Zhao Jin: assuming \( G \) is also acyclic, then
   \[
   \text{radius}(G) = \lceil \frac{\text{diameter}(G)}{2} \rceil.
   \]  
   (1)

   REMARKS: This shows that for acyclic connected bigraphs, you can reduce the computation of radius to the computation of diameter. But could you reduce the computation of diameter to radius?

   Recall that hw5 showed
   \[
   \text{radius}(G) \leq \text{diameter}(G) \leq 2 \text{radius}(G).
   \]  
   (2)

   Both inequalities in (2) are tight: for all \( n \) there are graphs \( H_n \) where \( \text{diameter}(H_n) = \text{radius}(H_n) = n \) and also graphs \( G_n \) where \( n = \text{diameter}(G_n) = 2 \text{radius}(G_n) \). Thus, (1) tells us that the graphs \( G_n \) cannot be acyclic.

2. (12+12 Points, \((a, b)\)-search trees)
   We consider the tradeoffs in using one of the following schemes to organize the nodes of an \((a, b)\)-search tree: (i) an array, (ii) a singly-linked list, (iii) a doubly-linked list, (iv) a balanced binary search tree.

   Consider a specific numerical example: block size is \( 4096 = 2^{12} \) bytes, and each block pointer is 4 bytes, and each key 6 bytes. A local pointer within the block uses 12 bits, but for simplicity treat this as two bytes. Please be sure to note other information you need in a node, such as a parent pointer, the degree, etc.

   (a) What is the maximum value of the parameter \( b \) under each of the schemes (i)-(iv)? Be sure to show your calculations. The lecture notes has some discussion of these issues.

   (b) What is the worst case time to search for a key in an \((a, b)\)-search tree with two million items? We need one number (not expression) for each of the four schemes as answer: the unit for each number is CPU cycles (or “CPUC”).

   MAKE THESE ASSUMPTIONS: Each disk I/O takes 1000 CPU cycles. If searching for a key takes \( O(\log n) \) or \( O(n) \) CPU time, always assume that “4” is the constant in big-Oh notation. E.g., searching for a key in a balanced BST with \( n = 100 \) keys takes \( 4 \log n = 4 \times 7 = 28 \) CPUC. Searching for in a list of length \( n = 100 \) takes \( 4n = 400 \) CPUC. The root of the search tree is always in main memory, so you never need to read or write the root. Assume \( a = \lfloor (b + 1)/2 \rfloor \).

   You can use calculators, but I encourage you to make calculator-free simplified estimates whenever possible (e.g., log base 2 of one million is 20).

3. (0 Points – do not hand in)
   Do part (b) of the previous problem, but for the Insert and Delete operations.

4. (5 Points, General Bin Packing)
   Recall that the general bin packing problem can be solved in time to \( O((n/e)^n + (1/2)) \) (see Lemma 2 of Lect.V, p.4) We ask you to improve this to \( O((n/e)^{n-(1/2)}) \).

   HINT: Repeat the trick which saved us a factor of \( n \) in the first place. Fix two weights \( w_1, w_2 \). We need to consider two cases: either \( w_1, w_2 \) belong to the same bin or they do not.
5. (3 Points, 2-Car Loading Policies)

We consider two policies for loading a front and rear car. Let us introduce a useful concept: if the total weight of the riders loaded in a car is $W$, then we say the car has residual capacity of $M - W$. Thus an empty car has residual capacity of $M$, and a completely full car has residual capacity of 0. If a new rider has weight at most the residual capacity of a car, then we say the rider fits into that car. Here are two possible loading policies for $G_2$:

- **First Fit Policy**: Load each rider into the front car if it fits, otherwise load into the rear car if it fits. If it fits neither car, dispatch the front car.
- **Best Fit Policy**: Load each rider into the car with the “best fit”, i.e., the car with the minimum residual capacity that fits the rider. If it fits neither car, then dispatch the front car.

Let $G_2(w)$ and $G_2'(w)$ denote (respectively) the number of cars used when loading according to the First Fit and Best Fit Policies. Note that the text describes only the First Fit Policy, and proves that $G_2(w) \leq G_1(w)$.

(a) Show an example where $G_2(w) > G_2'(w)$.
(b) Show an example where $G_2(w) < G_2'(w)$.

6. (8 Points, Sorted linear bin packing)

Suppose the input $w = (w_1, \ldots, w_n)$ is sorted as follows: $w_1 \geq w_2 \geq \cdots \geq w_n$.

(a) Prove that $G_1(w) \leq 1.5Opt(w)$ where $Opt(w)$ is the optimal bound for general bin packing.
(b) Give examples showing that this factor of 1.5 is the best possible.

7. (2 Point, Coin Change Problem)

Fix a currency system $D = (1, d_2, \ldots, d_m)$. Recall the basic definitions for coin changing problems found in p.7 (Lecture V). For any $x$ let $G(x) = (s_1, \ldots, s_m)$ be the greedy solution and $Opt(x) = (s_1^*, \ldots, s_m^*)$ is the optimum solution. Suppose $x$ is the smallest counter example to the canonicity of $D$. Show that

$$s_i \cdot s_i^* = 0 \quad \text{(for all } i = 1, \ldots, m)$$

8. (5+0+2 Points, Coin Changing Problem)

How do you prove that the US currency system is canonical? We will provide an inductive approach here.

REMARK: Part(b) seems hard, and is considered extra credit work. DO not hand it in unless you have a good solution.

Suppose $D = (1, d_2, \ldots, d_m)$ is a canonical currency system ($m \geq 1$). We look at extensions of $D$:

(a) If $D' = (1, d_2, \ldots, d_m, d_{m+1})$ extends $D$ with a denomination $d_{m+1} := cq_m$ for some $q \geq 2$, then $D'$ is called a Type A extension of $D$. Show that Type A extensions of $D$ are canonical.
(b) Assume $D'$ is a Type A extension of $D$ as in part(a). Let $D'' = (1, d_2, \ldots, d_m, d_{m+1}, d_{m+2})$ extend $D'$ with a denomination $d_{m+2} = ad_{m+1} + bd_m$ where $a, b$ are non-negative integers. If $q \leq 3$ and $a \geq 1$ then $D''$ is called a Type B extension of $D'$. Show that Type B extensions of a Type A extension $D'$ is canonical.

Example: let $D = (1, 5)$, which is canonical. Then $D' = (1, 5, 10)$ is a Type A extension of $D$. Then $D'' = (1, 5, 10, 25)$ is a Type B extension of $D'$.
(c) Conclude that the US currency system comprising the notes $\$100, $\$50, $\$20, $\$10, $\$5$ and coins $\$25e, $\$10e, $\$5e, $\$1e$ is canonical.

9. (8 Points, Sorted linear bin packing)

(a) Show a currency system that is complete and canonical but does not have uniqueness. HINT: you need not consider more than 3 denominations.
(b) Show that the binary system $D = (1, 2, 4, \ldots, 2^m)$ is a canonical system that is also unique.

10. (3+4 Points)

We gave four different greedy criteria for the activities selection problem. Be sure to make your examples as small and simple as possible (we will take off points for unnecessary complications).

(a) Show that the other three criteria are suboptimal.
(b) Actually, each of the four criteria has an inverted version in which we sort in decreasing order (break ties arbitrarily). Show that each of these inverted criteria are also suboptimal.